



Estimation of a semi-physical GLBE model using dual EnKF learning algorithm coupled with a sensor network design strategy: application to air field monitoring

Gilles Roussel, Laurent Bourgois, Mohammed Benjelloun, Gilles Delmaire

► To cite this version:

Gilles Roussel, Laurent Bourgois, Mohammed Benjelloun, Gilles Delmaire. Estimation of a semi-physical GLBE model using dual EnKF learning algorithm coupled with a sensor network design strategy: application to air field monitoring. Information Fusion, 2013, 14, pp.335-348. 10.1016/j.inffus.2013.03.001 . hal-00862593

HAL Id: hal-00862593

<https://hal-centralesupelec.archives-ouvertes.fr/hal-00862593>

Submitted on 17 Sep 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Estimation of a semi-physical GLBE model using dual EnKF learning algorithm coupled with a sensor network design strategy: application to air field monitoring

Gilles Roussel^{a,*}, Laurent Bourgois^b, Mohammed Benjelloun^a, Gilles Delmaire^a

^a*LISIC, 50 rue Ferdinand Buisson, 62228 Calais Cedex, France*

^b*SUPELEC ESS, 3 rue Joliot-Curie, 91192 Gif-sur-Yvette Cedex, France*

Abstract

In this paper, we present the fusion of two complementary approaches for modeling and monitoring the spatio-temporal behavior of a fluid flow system. We also propose a mobile sensor deployment strategy to produce the most accurate estimate of the true system state. For this purpose, deterministic and statistical information was used. We adopted a filtering method based on a semi-physical model which derives from a fluid flow numerical model known as lattice Boltzmann model (LBM). The *a priori* physical knowledge was introduced by the Navier-Stokes equations which were discretized by the lattice Boltzmann approach. Moreover, its multiple-relaxation-time (MRT) variant not only improved the stability, but also enabled the introduction of additional degrees of freedom to be estimated like the synaptic weights of a neural network. The statistical knowledge was then introduced into the model by performing a sequential learning of these parameters and an estimation of the speed field of the fluid flow starting from measurements. The low spatial density of measurements, the large amount of data inherent to environmental issues and the nonlinearity of the generalized lattice Boltzmann equations (GLBE) enjoined us to use the ensemble Kalman filter (EnKF) for the recursive estimation procedure. A dual state-parameter estimation which results in a significantly reduced

*Corresponding author. Tel.: +33 321 463 697; Fax: +33 321 460 686.

Email address: gilles.roussel@lisic.univ-littoral.fr (Gilles Roussel)

computation time was used by combining two filters consecutively activated in the same iteration. Finally, we proposed to complete the lack of spatial information of the sparse-observation network by adding a mobile sensor, which was routed to the location where the cell-by-cell output estimation error was the highest. Experimental results in the context of the standard lid-driven cavity problem revealed the presence of few zones of interest, where fixed sensors can be deployed to increase performances in terms of convergence speed and estimation quality. Finally, the study showed the feasibility of introducing some additional parameters which act as degrees of freedom, to perform large-eddy simulation of turbulent flows without numerical instabilities.

Keywords:

Grey-box modeling, model fusion, sensor network design, air field monitoring

1. Introduction

This study aims to investigate spatio-temporal monitoring of fluid flow behavior by collecting data from a sensor network. In this kind of problem, flow and concentration fields are usually computed using data assimilation techniques. However, typical issues such as modeling, learning and sensor network design are regularly encountered. Generally, the model based on physical knowledge is supposed to be sufficiently refined and the spatial description of the study area is well defined. When experimental data are approximate and uncertainties result from the lack of sufficient physical knowledge, it is preferable to use an adaptive model. We proposed to build an adaptive model by merging the deterministic knowledge of physics with the statistical knowledge of filtered data. The main difference lay in the fact that the constructed model could be considered as a semi-physical model. Indeed, new degrees of freedom were added and the model was able to adapt to different kinds of driving behavior with the help of statistical methods. Our approach consisted in a state-space filtering problem with the objective of learning the system parameters of a conventional computational fluid dynamic (CFD) model. The *a priori* knowledge of the physical

laws that govern the studied system was introduced by a Navier-Stokes model discretized by the lattice Boltzmann approach for fluid flow simulation [1], [2], [3], [4], [5]. This deterministic model which operated in the forward direction, was able to reproduce the macroscopic behavior of a fluid accurately by a microscopic simulation of particle dynamics. Its alternative multiple-relaxation-time form did not only improve the stability of the method [6], [7] but also allowed the introduction of additional degrees of freedom which had to be identified like the synaptic weights of a neural network. Local knowledge was then statistically inducted into the model by performing a sequential learning of these new parameters and an estimation of the fluid flow velocity field directly from measurements of the real environment. Our goal was to develop an efficient method to incorporate measures so as to ensure that the semi-physical model could adapt to the actual behavior of the system. The low spatial density of measurements, the large amount of data inherent to environmental issues and the nonlinearity of the LBM led us to use a sequential state-parameter estimation technique based on the EnKF algorithm [8], [9], [10]. For that purpose, a state-space representation of the GLBE has been defined for the recursive estimation procedure [11]. Besides, to estimate both uncertain parameters and state variables conjointly, a dual formulation based on the addition of a second filter running simultaneously was proposed. This technique reduced the computational cost in the presence of a large number of parameters compared to a conventional joint estimation method [12]. Finally, to estimate the total number of variables in a suitably way, the lack of available information also had to be considered. Consequently, a sensor network design strategy was developed to collect relevant information on a given environment. The idea consisted in adapting the observatory network in function of the cell-by-cell output estimation error calculations. The lack of spatial information of a sparse-observation network was completed by adding a mobile sensor which moved to the position where the value of the analysis output error covariance matrix was high.

The structure of the article is the following. In section 2, the lattice Boltzmann model is derived under its generalized form and a mathematical standard

state-space representation is defined in section 3. In section 4, the dual state-parameter ensemble Kalman filter algorithm is studied and a compatible sensor network deployment strategy is detailed. Then our proposed model fusion approach is applied in the case of lid-driven cavity flow in section 5. Finally, concluding remarks and perspectives are given in section 6.

2. Lattice Boltzmann model

2.1. General framework

The LBM is a numerical method based on Boltzmann kinetic theory and can be expressed in terms of the probability to find a fluid particle, in the vicinity of a given location and time, that is moving in one of a number of discrete directions. The particular structure considered in this work is the so called D2Q9 model, which is a two-dimensional lattice with nine links representing nine velocity vectors (including the zero velocity). In this model [1], space is discretized into square lattice whose spatial step is δ_x . The time step δ_t was chosen so that the unit of velocity $c = \delta_x/\delta_t$ remained constant during the whole study. Particles move along the links of the square lattice from one node of the grid to one of its neighbors as illustrated by the nine vectors in figure 1 and given by:

$$\mathbf{e}_\alpha = \begin{cases} (0, 0)^t, & \alpha = 0 \\ \left(\cos \left[(\alpha - 1) \frac{\pi}{2} \right], \sin \left[(\alpha - 1) \frac{\pi}{2} \right] \right)^t, & \alpha = 1, \dots, 4 \\ \left(\cos \left[(2\alpha - 9) \frac{\pi}{4} \right], \sin \left[(2\alpha - 9) \frac{\pi}{4} \right] \right)^t, & \alpha = 5, \dots, 8 \end{cases} \quad (1)$$

The nine discrete velocities were given by:

$$\boldsymbol{\xi}_\alpha = \mathbf{e}_\alpha \frac{\delta_x}{\delta_t}, \quad \alpha = 0, \dots, 8 \quad (2)$$

The single-particle velocity distribution function along the α direction at a particular discrete time t and at location \mathbf{r} was denoted as $f_\alpha(\mathbf{r}, t)$. These

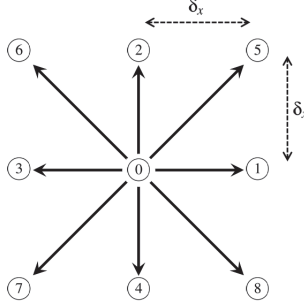


Figure 1: Vectors along the links of the square lattice, as used in D2Q9 LBM.

distribution functions evolved according to a Boltzmann equation that is discrete in both space and time:

$$f_\alpha(\mathbf{r} + \boldsymbol{\xi}_\alpha \delta_t, t + \delta_t) = f_\alpha(\mathbf{r}, t) + \Omega(f_\alpha(\mathbf{r}, t)) \quad (3)$$

The evolution consisted of two fundamental steps: the advection step (motion to the relevant neighbors) and the collision step (redistribution of the distribution at each node). The most convenient choice for the collision operator Ω was a single time relaxation form based on the BGK approximation [13], *i.e.* by applying a linearization around an equilibrium distribution function f_α^{eq} defined with respect to the conservation laws:

$$\Omega(f_\alpha) = -\frac{\delta_t}{\lambda} (f_\alpha - f_\alpha^{eq}) \quad (4)$$

The dimensionless relaxation time λ characterizes the return to equilibrium and must satisfy the stability condition of the explicit Euler scheme:

$$0 \leq \frac{\delta_t}{\lambda} \leq 2 \quad (5)$$

It is advised to get rid of the null value, since otherwise it would mean that the equilibrium state has already been reached. One commonly denotes by $\tau = \lambda/\delta_t$ the dimensionless relaxation time. For an athermal medium, the conserved macroscopic quantities are the density ρ and the momentum $\mathbf{j} = \rho \mathbf{u}$ where \mathbf{u} is the fluid velocity (the energy is not considered as a conserved quantity). These

conserved quantities are related to the distributions by:

$$\left\{ \begin{array}{l} \rho = \sum_{\alpha=0}^8 f_{\alpha} = \sum_{\alpha=0}^8 f_{\alpha}^{eq} \\ \mathbf{j} = \sum_{\alpha=0}^8 \boldsymbol{\xi}_{\alpha} f_{\alpha} = \sum_{\alpha=0}^8 \boldsymbol{\xi}_{\alpha} f_{\alpha}^{eq} \end{array} \right. \quad (6)$$

Finally, the equilibrium distribution functions may be computed at each time step for each node from the conserved macroscopic quantities:

$$f_{\alpha}^{eq} = w_{\alpha} \rho \left(1 + \frac{\boldsymbol{\xi}_{\alpha} \cdot \mathbf{u}}{c_s^2} + \frac{1}{2} \left(\frac{\boldsymbol{\xi}_{\alpha} \cdot \mathbf{u}}{c_s^2} \right)^2 - \frac{\mathbf{u}^2}{2c_s^2} \right) \quad (7)$$

In this suitable formulation [2, 1], the speed of sound c_s and the weighting factors w_{α} depend on the lattice geometry. For the D2Q9 model, the speed of sound is defined as $c_s^2 = c^2/3$ and the weighting factors are given by:

$$w_{\alpha} = \left\{ \begin{array}{ll} 4/9, & \alpha = 0 \\ 1/9, & \alpha = 1, \dots, 4 \\ 1/36, & \alpha = 5, \dots, 8 \end{array} \right. \quad (8)$$

Figure 2 gives an overview of the stream and collide steps for a fluid cell during one time step.

Sometimes, for a given direction $\alpha \neq 0$, the corresponding neighbor cell may be solid. In this case, the unknown incoming distribution function along the direction α had to be constructed by setting artificial boundary conditions. The no-slip (or homogeneous Dirichlet) boundary condition which required that the normal and tangential components of the fluid velocity along the edge of the obstacle be zero, was approximated using the standard bounce-back rule. Concretely, the known outgoing distribution function pointing into the solid re-enters the grid at the same node, but associated with the opposite direction $\tilde{\alpha}$. If the obstacle is moving, fluid particles moving across the solid cells have to be accelerated. This can be done by adding an artificial forcing term, which depends on the velocity of the obstacle, during streaming. This non-homogeneous Dirichlet boundary condition is discussed in detail in [14, 15].

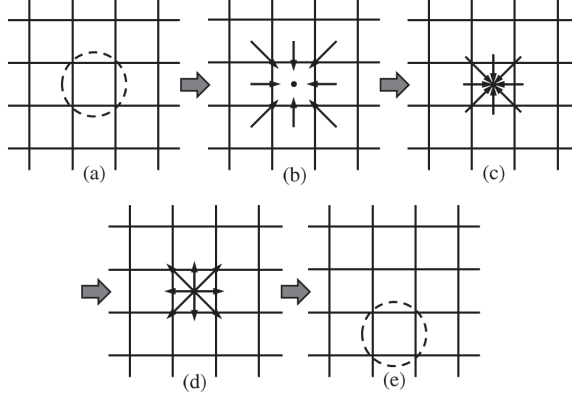


Figure 2: *Advection-collision process for a fluid cell: (a) Select fluid cell to treat, (b) Stream f_α from the relevant neighbors, (c) Full set of streamed f_α , (d) Collide streamed f_α by computing ρ , \mathbf{j} and f_α^{eq} , (e) Store new f_α in target grid and continue with next cell.*

The classical Navier-Stokes equations can be recovered from the macroscopic lattice Boltzmann equations by a Chapman-Enskog expansion procedure [3]. Moreover, other physical quantities as the pressure $p = c_s^2 \rho$ directly depend on the density [4, 5] and the kinematic viscosity ν can be obtained from the relaxation time as follows:

$$\nu = c_s^2 \delta_t \left(\frac{\lambda}{\delta_t} - \frac{1}{2} \right) \quad (9)$$

2.2. Smagorinsky subgrid model

In the case of turbulent flows, *i.e.* high Reynolds numbers, the kinematic viscosity value decreases which implies decreasing λ/δ_t toward 1/2 (see equation (9) above). Unfortunately, considering (5), a value of δ_t/λ close to 2 leads to numerical instabilities. One solution proposed in [16] to perform large-eddy simulation (LES) of turbulent flows consists in using the Smagorinsky subgrid model [17]. Note that such subgrid model have nothing to do with grid refinement. Its primary use is to represent the influence of the unresolved small scales (below the lattice spacing) by locally adding a positive turbulent (or eddy) viscosity ν_t to the kinematic shear viscosity so that $\nu^* = \nu + \nu_t$. Concretely, the single-relaxation-time approximation is abandoned and a new spatially-dependent di-

mensionless relaxation time τ^* is computed at each time step as follows (see [4] for details):

$$\tau^* = \frac{1}{2} \left(\tau + \sqrt{\tau^2 + \frac{2C_{Smago} \delta_x^2}{\rho_0 c_s^4 \delta_t^2} |\Pi^{neq}|} \right) \quad (10)$$

Here, C_{Smago} is known as the Smagorinsky constant whose value is empirically obtained, ρ_0 denotes the mean density and Π^{neq} corresponds to the non-equilibrium stress tensor where:

$$|\Pi^{neq}| = \sqrt{\sum_{i,j} \Pi_{ij}^{neq} \Pi_{ij}^{neq}} \quad (11)$$

The subscript indices i and j had been used to identify the direction in Cartesian space and Π_{ij}^{neq} could be calculated for a given node only using neighboring nodes from the following relation where the notation $\xi_{\alpha i}$ denoted the i^{th} Cartesian component of the vector ξ_{α} :

$$\Pi_{ij}^{neq} = \sum_{\alpha=0}^8 \xi_{\alpha i} \xi_{\alpha j} (f_{\alpha} - f_{\alpha}^{eq}) \quad (12)$$

2.3. Generalized lattice Boltzmann equation

The GLBE introduced by d'Humières [6, 7] was used in this work. Due to its flexibility in using disparate relaxation times, the GLBE provided numerical stability at coarse grids. The idea consisted in associating the vector of velocity distribution functions $\mathbf{f} = (f_0, \dots, f_8)^t$ to the vector of moments $\mathbf{m} = (m_0, \dots, m_8)^t$. Each moment m_{α} was defined as a linear combination of velocity distribution functions. Consequently, there existed a linear transformation matrix $M \in \mathcal{M}_9(\mathbb{R})$ such as $\mathbf{m} = M\mathbf{f}$ which transformed a vector in the vector space spanned by the discrete velocities into a vector in the vector space

spanned by the moments:

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 0 & c & 0 & -c & 0 & c & -c & -c & c \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & c & 0 & -c & c & c & -c & -c \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \end{bmatrix} \quad (13)$$

In this manner, the nine components of the vector \mathbf{m} have a physical interpretation: $m_0 = \rho$ is the density, $m_1 = e$ is related to the kinetic energy, $m_2 = \chi$ is related to the kinetic energy square, $m_3 = j_x$ and $m_5 = j_y$ are the x and y components of the momentum, $m_4 = q_x$ and $m_6 = q_y$ are proportional to the x and y components of the energy flux, and $m_7 = \Pi_{xx}$ and $m_8 = \Pi_{xy}$ corresponded to the diagonal and off-diagonal components of the viscous stress tensor. Explicitly, these moments were constructed by applying the Gram-Schmidt orthogonalization procedure to polynomials of Cartesian components of the discrete velocities [7, 18]. The matrix M thus constructed is naturally an orthogonal matrix invertible. Besides, it is interesting to note that m_0 , m_3 and m_5 correspond to the conserved macroscopic quantities of (6). Thus, the collision operator Ω is now described in moment space:

$$\Omega(m_\alpha) = -\frac{\delta_t}{\lambda_\alpha} (m_\alpha - m_\alpha^{eq}) \quad (14)$$

As a result, the non-conserved moments relaxed linearly towards an equilibrium state according to simple relaxation equations with constant relaxation times λ_α for $\alpha = \{0, \dots, 8\}$. Obviously, the conserved moments were not affected by collisions since $m_\alpha = m_\alpha^{eq}$ for $\alpha = \{0, 3, 5\}$. Consequently, the actual values of the relaxation parameters $s_\alpha = \delta_t/\lambda_\alpha$ for conserved moments were irrelevant and set to zero in what followed. The stability condition of the explicit Euler scheme

must be satisfied for the non-conserved moments, *i.e.* for $\alpha = \{1, 2, 4, 6, 7, 8\}$:

$$0 < s_\alpha \leq 2 \quad (15)$$

With the particular relaxation parameter vector $\mathbf{s} = (0, s_1, s_2, 0, s_4, 0, s_6, s_7, s_8)^t$ and the corresponding diagonal relaxation matrix $S = \text{diag}(\mathbf{s}) \in \mathcal{M}_9(\mathbb{R})$, the Boltzmann equation became:

$$\mathbf{f}(\mathbf{r} + \boldsymbol{\xi}\delta_t, t + \delta_t) = \mathbf{f}(\mathbf{r}, t) - M^{-1}S[\mathbf{m}(\mathbf{r}, t) - \mathbf{m}^{eq}(\mathbf{r}, t)] \quad (16)$$

In this new Boltzmann equation, the collision step was executed in the moment space, while the advection step was performed in the distribution function space. The column vector $\mathbf{f}(\mathbf{r} + \boldsymbol{\xi}\delta_t, t + \delta_t)$ is defined by:

$$\mathbf{f}(\mathbf{r} + \boldsymbol{\xi}\delta_t, t) = (f_0(\mathbf{r} + \boldsymbol{\xi}_0\delta_t, t), \dots, f_8(\mathbf{r} + \boldsymbol{\xi}_8\delta_t, t))^t \quad (17)$$

The equilibrium values of the non-conserved moments in the above equations were nonlinear polynomial functions of the conserved quantities whose expression values were obtained by computing the moments of the continuous Maxwell-Boltzmann velocity distribution function. This choice of parameters respected symmetry arguments, yielded Galilean invariance and optimized isotropy and stability of the model. See [7] for the complete derivation of these properties. Besides, the authors also recommend fixing $s_4 = s_6$ when the no-slip boundary condition is applied and $s_7 = s_8$ to enforce isotropy. Finally, the shear viscosity ν and the bulk viscosity ς of the model are given by:

$$\nu = \frac{1}{3} \left(\frac{1}{s_7} - \frac{1}{2} \right) c^2 \delta_t \quad (18)$$

and:

$$\varsigma = \frac{2}{9} \left(\frac{1}{s_1} - \frac{1}{2} \right) c^2 \delta_t \quad (19)$$

Thus, a distinct advantage of the MRT model was the possibility to independently control the kinematic shear and bulk viscosities with the help of relaxation parameters. Consequently, only s_2 , s_4 and s_6 had no physical significance but remained adjustable to improve stability. The values of these unknown parameters could be determined by applying the standard von Neumann stability analysis to the linearized Boltzmann equation [7].

3. Nonlinear state-space representation

Our goal is to make the mathematical representation of the LBM compatible with a recursive estimation procedure. Concretely, we wished to convert the MRT model to a state-space representation which is perfectly adapted to the EnKF method. By definition, state variables summarize the effects of past decisions and describe the system at a given time. They constitute the components of a state vector whose time sequence characterizes system dynamics and contributes to forecast future behavior. Consequently, the LBM state-space representation is spontaneously obtained by assigning to the moments of each cell the role of state variable. Moreover, in the MRT approach, some relaxation parameters have no physical meaning and may assume the role of additional degrees of freedom.

3.1. Moment space description

Starting from the definition of the GLBE approach, and after a slightly rewriting work, the MRT evolution equation (16) can be modified to get:

$$m_\alpha(\mathbf{r}, t + \delta_t) = \sum_{i,j=0}^8 M_{\alpha i} M_{ij}^{-1} [(1 - s_j) m_j(\mathbf{r} - \boldsymbol{\xi}_i \delta_t, t) + s_j m_j^{eq}(\mathbf{r} - \boldsymbol{\xi}_i \delta_t, t)] \quad (20)$$

which can be rewritten in the following concise vector form:

$$\mathbf{m}(\mathbf{r}, t + \delta_t) = M \text{diag} (M^{-1} [(I_9 - S) V(\mathbf{r} - \boldsymbol{\xi} \delta_t, t) + S V^{eq}(\mathbf{r} - \boldsymbol{\xi} \delta_t, t)]) \quad (21)$$

where I_9 is the identity matrix of order 9 and $V(\mathbf{r} - \boldsymbol{\xi} \delta_t, t)$ and $V^{eq}(\mathbf{r} - \boldsymbol{\xi} \delta_t, t)$ are two square matrices of order 9, which are given by:

$$V(\mathbf{r} - \boldsymbol{\xi} \delta_t, t) = [\mathbf{m}(\mathbf{r} - \boldsymbol{\xi}_0 \delta_t, t) \dots \mathbf{m}(\mathbf{r} - \boldsymbol{\xi}_8 \delta_t, t)] \quad (22)$$

and:

$$V^{eq}(\mathbf{r} - \boldsymbol{\xi} \delta_t, t) = [\mathbf{m}^{eq}(\mathbf{r} - \boldsymbol{\xi}_0 \delta_t, t) \dots \mathbf{m}^{eq}(\mathbf{r} - \boldsymbol{\xi}_8 \delta_t, t)] \quad (23)$$

As we will see later on, it may be judicious to consider that the relaxation parameters differ spatially. This possibility is simply anticipated by considering $S(\mathbf{r})$ instead of S in (21).

Finally, by remembering that for each location \mathbf{r} of the grid at time t , the equilibrium moments $m_\alpha^{eq}(\mathbf{r}, t)$ are nonlinear polynomial functions of the conserved moments $m_0(\mathbf{r}, t)$, $m_3(\mathbf{r}, t)$ and $m_5(\mathbf{r}, t)$ [7], we have succeeded in entirely expressing the lattice Boltzmann evolution equation in the vector space spanned by the moments. However, such a formulation requires the knowledge of all the moments from all the neighbors which increases the computational cost.

Besides, as we have previously exposed, each component of the moment vector has a physical interpretation. In particular, the components m_0 , m_3 and m_5 stand for density and x and y components of the momentum respectively can be directly used to compute the x and y components of the fluid velocity as follows:

$$\mathbf{u} = \begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} j_x/\rho \\ j_y/\rho \end{bmatrix} = \begin{bmatrix} m_3/m_0 \\ m_5/m_0 \end{bmatrix} \quad (24)$$

In this manner, the fluid velocity can be also locally expressed with respect to the moment vector.

3.2. Towards one multi-dimensional formulation

From the previous relation (21), we observed that the temporal evolution of the moment vector on a lattice site \mathbf{r} did not only depend on the moment vector on this particular site, but also (and highly) on the moment vectors of its neighborhood. This multi-dimensional representation of the moment variables could be expressed as vector form, which was more adapted to state-space analysis. Thus all the moment vectors of the domain were concatenated in a single column vector $\underline{\mathbf{m}} \in \mathbb{R}^{9N_n}$, which represented the state of the system, and where $N_n = N_x \times N_y$ was the total number of nodes. This step is graphically illustrated in figure 3 and from now on, each underlying vector corresponds to the concatenation of the N_n corresponding vectors of the domain.

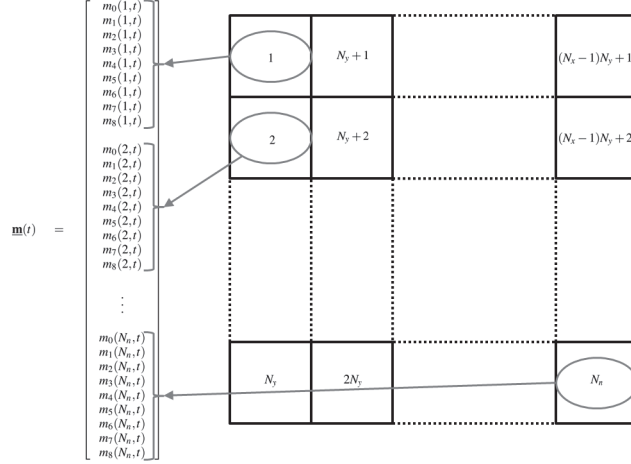


Figure 3: Concatenation step of all the moment vectors of the domain.

From then on, an exhaustive description of the whole domain, *i.e.* the perfect knowledge of each solid cell position (and the corresponding boundary conditions), enabled us to write:

$$\underline{\mathbf{m}}(t + \delta_t) = g(\underline{\mathbf{m}}(t), \mathbf{s}) \quad (25)$$

where g is the state-space forward propagator. In the particular case where the relaxation parameters are spatially heterogeneous, the vector of relaxation parameters over the entire domain $\underline{\mathbf{s}}$ must be considered instead of \mathbf{s} in (25).

By following the same line and defining the nonlinear vector function h , we could generalize (24) and obtained the output equation (26), where $\underline{\mathbf{u}} \in \mathbb{R}^{2N_n}$ stands for the fluid velocity over the entire domain.

$$\underline{\mathbf{u}}(t + \delta_t) = h(\underline{\mathbf{m}}(t + \delta_t)) \quad (26)$$

3.3. Change of notation

In order to relieve the state-space representation and to use a typical control engineering formalism, we applied a change of notation. The vector of moments over the entire domain which represented the state of the system was from then on noted \mathbf{x} and denoted the state vector. The vector of relaxation parameters

over the entire domain simply became the vector of parameters $\boldsymbol{\theta}$ and the vector of fluid velocity over the entire domain changes into \mathbf{y} the output vector. Finally the discrete time was represented as subscript indices and we replaced $t + \delta_t$ by $t+1$. Consequently, we could write the following nonlinear discrete time system:

$$\begin{cases} \mathbf{x}_{t+1} &= g(\mathbf{x}_t, \boldsymbol{\theta}) \\ \mathbf{y}_{t+1} &= h(\mathbf{x}_{t+1}) \end{cases} \quad (27)$$

Let us underline the distinction between the vector of fluid velocity over the entire domain $\mathbf{u} \in \mathbb{R}^{2N_n}$ and the observation vector \mathbf{y} . Indeed, the expert does not automatically have measurements at each lattice site, but only at sensor locations. Concretely, if each sensor measures the x and the y component of the fluid velocity and if N_c denotes the total number of available sensors, we have $\mathbf{y} \in \mathbb{R}^{2N_c}$.

In the case where an input command \mathcal{U}_t must be taken into account, *e.g.* in the case of forced flux, the system (27) becomes:

$$\begin{cases} \mathbf{x}_{t+1} &= g(\mathbf{x}_t, \boldsymbol{\theta}, \mathcal{U}_t) \\ \mathbf{y}_{t+1} &= h(\mathbf{x}_{t+1}) \end{cases} \quad (28)$$

4. Dual ensemble Kalman filter for data assimilation

4.1. State of the art

In practice, physical data assimilation applied to environmental surveillance encounters many difficulties. By excluding variational techniques, we turned our attention to sequential filtering techniques based on a prediction-correction scheme. Most difficulties in filtering algorithms design are often caused by the nature of the evolution model. The spatio-temporal evolution of high-dimensional nonlinear systems, in presence of real-time estimation constraints, fast-sampling constraints (which ensure numerical stability) and uncertainties, make the optimization of existing filtering methods indispensable. The dual EnKF which simultaneously optimizes model parameters and state variables is a recursive method suitable for this kind of problem.

Many approaches, which combined both parameter identification and state estimation of a dynamic system, proceed with joint estimation, *i.e.* where the unknown state and parameter vectors were concatenated to form the new augmented state vector [19, 20]. It was preferable to separate the state and parameter estimations for high-dimensional problems in order to reduce the length of the vector and the size of matrices. This approach consisted in combining two filters consecutively activated in the same iteration to estimate first, the parameters from the last updated state and, second, the new state from the parameters previously analyzed. The procedure did not require any order of priority between the state and parameter estimations. This kind of filtering can be used to estimate most uncertain sources, even for time-varying parameters. This approach was initially applied to a standard Kalman Filter (KF) in the context of hydrology [21, 22].

If the process to be estimated and (or) the measurement relationship to the process were nonlinear, a modified version of the Kalman filter was required. Thus, some examples of dual estimation were studied with the extended Kalman filter (EKF) [23], especially for neural model estimation [24, 25], where the state-propagation Jacobian matrix was evaluated at each time step around the current estimate and used in the Kalman filter equations. In case of high nonlinearity, both prior and posterior probability distributions were not Gaussian anymore. As a result, the first and second moments are not sufficient to characterize the entire state distribution. This approximation often led to instability or divergence [26, 8, 27, 28].

The unscented Kalman filter (UKF), proposed by Julier and Uhlmann, permits to enhance the robustness to highly nonlinear systems without significantly increasing the computational complexity [29]. In addition, this technique does not require the derivation of any Jacobians. The UKF uses a deterministic sampling technique to pick a minimal set of sample points, called sigma points, whose number depends on the length of the state vector. These sigma points are then propagated through the nonlinear system and capture the posterior mean and covariance accurately to the second order for any nonlinearity, while

minimizing the third order (Taylor series expansion).

The computational complexity of both EKF and UKF algorithms is cubic, *i.e.* $O(n^3)$ where n is the dimension of the state vector, due to the covariance matrix update. Unfortunately, this computation quickly becomes intractable in practice for large state dimension.

An alternative to overcome this drawback was to use a suboptimal estimator. The EnKF, proposed by Evensen, is a suboptimal particle filter of stochastic nature based on the Monte Carlo simulation technique [8, 9, 10, 30, 31, 32]. The fitness for the estimation of a high dimensional vector of a nonlinear model with a reduced dimension vector observation led to use this method. Furthermore, the method gave good results for the joint estimation of state variables and parameters. It is actually recognized in [33] that this method approximates the probability density function (pdf) by a set of discrete particles, typically 50 to 100 is sufficient, hence making feasible implementation of the EnKF in situations where the forward step of the data assimilation is computationally expensive. With EnKF, the *a posteriori* pdf of the exact particle filter method was approximated to a simple Gaussian estimation with only the first two moments. For linear model, overall integration was consistent with integration of the exact equation error covariance within the limits of a set of infinite size. Obviously the method was sub-optimal for nonlinear system, due to a lower estimate of the *a posteriori* probability density, which could lead to divergence in some rare cases, although the number of achievements was endless. However, the literature suggests some localization methods to reduce the impact of measures located very far from the prediction point. Newer versions of estimation Monte Carlo algorithm are as fast as EnKF and have an asymptotic behavior as robust as the particle filter based on the estimation of a true posterior probability density by a Gaussian mixture [34]. Although its use depends on the choice of two important parameters, the adaptive parameter bandwidth α and mixing parameter setting h , this adaptive filter Gaussian mixture could quite suitable for a more robust estimate for data assimilation.

The EnKF principle consists in evaluating the empirical covariance matrix

of an ensemble of possible state vectors randomly generated from the current state, instead of computing its exact value by propagation through the model. By assuming a Gaussian distribution of state and measurements, it was possible to apply a linear correction procedure. As a result, the computational complexity was linear, *i.e.* $O(m^3)$ where m was the product between the dimension of the state vector, the number of measurements and the number of samples per iteration. The computational cost was largely lower than the one obtained by the previous filters. For example, let us consider the estimation of pollutant concentration field in a square area of $1\text{km} \times 1\text{km}$. For a spatial step of 1m, 10 measurements and 50 Monte Carlo samples per iteration, the EnKF algorithmic cost is $O(5.0 \times 10^8)$ versus $O(10^{18})$ for the EKF and UKF. Thus, the EnKF is capable to estimate a state vector with several thousands of variables, while operating near the Cramer-Rao lower bound [35, 36, 37]. The EnKF approach is also naturally parallelizable since each ensemble member can be updated independently. Despite some resemblances to particle filters, the EnKF belongs to the Gaussian filter family due to its limitation to second-order statistics. Nevertheless, this kind of filtering provides proper estimation of the state with a relatively low number of samples compared to particle filtering. Moreover, the EnKF gives better results compared to variational methods when the dimension of the state vector is large. Indeed, it requires neither a derivation of the adjoint equations nor a retrograde integration of the evolution model. Finally, let us underline that a dual state-parameter estimation approach based on the ensemble Kalman filter in the case of constant parameters with a spatially dense observatory network was presented in [12].

4.2. General framework and notations

The EnKF is a sequential Monte Carlo method which empirically expresses the statistical properties of the estimator. For that purpose, the forecast error covariance matrix at time $t + 1$ is not computed anymore by propagating in time the analysis error covariance matrix at time t , but by propagating a finite ensemble of possible state vectors, which are randomly generated. The

error statistics, which is represented graphically by a cloud of dots, is propagated by the model without applying any linearization step. In what follows we considered a nonlinear discrete time system of the form:

$$\mathbf{x}_{t+1} = g(\mathbf{x}_t, \boldsymbol{\theta}, \mathcal{U}_t) + \mathbf{w}_t \quad (29)$$

In this system, the state-space forward propagator g is a nonlinear vector function which does not only depend on the state vector \mathbf{x}_t , but also on a hypothetical input command \mathcal{U}_t (forcing term) and sometimes on a vector of model parameters $\boldsymbol{\theta}$. The process noise \mathbf{w}_t is assumed to be additive Gaussian, with zero mean and with covariance matrix Q :

$$\mathbf{w}_t \sim \mathcal{N}(0, Q) \quad (30)$$

The nonlinear function h in the measurement equation (31) relates the state to the vector of output observations \mathbf{y}_{t+1} available at time $t + 1$.

$$\mathbf{y}_{t+1} = h(\mathbf{x}_{t+1}) + \mathbf{v}_{t+1} \quad (31)$$

All sources of errors in the observation are reflected by the vector \mathbf{v}_{t+1} , which is assumed to be Gaussian, with zero mean and with covariance matrix R :

$$\mathbf{v}_{t+1} \sim \mathcal{N}(0, R) \quad (32)$$

The process and measurement noises are assumed to be uncorrelated. The evolution mechanism of the EnKF is therefore based on a prediction-correction scheme.

During the prediction step, the forecast state ensemble is generated by reiterating N times (the number of ensemble members) the full nonlinear dynamical model (29). By this way, each particle explores the state space independently:

$$\mathbf{x}_{t+1}^{f,i} = g(\mathbf{x}_t^{a,i}, \boldsymbol{\theta}, \mathcal{U}_t) + \mathbf{w}_t^i, \quad i = 1, \dots, N \quad (33)$$

In this relation, the vector $\mathbf{x}_{t+1}^{f,i}$ represents the i^{th} forecast state ensemble member at time $t + 1$ and \mathbf{w}_t^i obviously indicates the associated process noise. Similarly, the vector $\mathbf{x}_t^{a,i}$ corresponds to the i^{th} analysis state ensemble member at

previous time t . The forecast output ensemble can be established as follow:

$$\mathbf{y}_{t+1}^{f,i} = h\left(\mathbf{x}_{t+1}^{f,i}\right), \quad i = 1, \dots, N \quad (34)$$

Here, the vector $\mathbf{y}_{t+1}^{f,i}$ indicates the i^{th} forecast output ensemble member at time $t + 1$. The forecast error covariance matrix \mathbf{P}_{t+1}^f at time $t + 1$ can be empirically estimated. Nevertheless, contrary to the Kalman filter, this calculation is optional. Indeed, this matrix does not involve in the evolution mechanism of the ensemble Kalman filter. However, the empirical estimation methodology is proposed in what follows for its role in another part of the algorithm. In a first time, the idea consists in estimating the true state of the system by calculating the ensemble mean $\bar{\mathbf{x}}_{t+1}^f$ at time $t + 1$:

$$\bar{\mathbf{x}}_{t+1}^f = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_{t+1}^{f,i} \quad (35)$$

The unbiased empirical estimate \hat{P}_{t+1}^f of the forecast error covariance matrix at time $t + 1$ was then obtained in a second time by the following equation:

$$\hat{P}_{t+1}^f = \frac{1}{N-1} \sum_{i=1}^N \left(\mathbf{x}_{t+1}^{f,i} - \bar{\mathbf{x}}_{t+1}^f \right) \left(\mathbf{x}_{t+1}^{f,i} - \bar{\mathbf{x}}_{t+1}^f \right)^t \quad (36)$$

In order to avoid the phenomenon of particle coalescence, Burgers and its co-authors showed in 1998 the necessity, when ensemble members were updated, to perturb observations and to empirically estimate the measurement error covariance matrix which appeared in the Kalman gain calculation [9, 10]. In this manner, the statistical sample of the estimates was constantly updated, which ensured sufficient particle diffusion and avoided divergence of the algorithm. The idea consisted in perturbing the current observation by adding a zero mean random variable with covariance matrix R , so that:

$$\mathbf{y}_{t+1}^i = \mathbf{y}_{t+1} + \mathbf{v}_{t+1}^i, \quad i = 1, \dots, N \quad (37)$$

The vector \mathbf{y}_{t+1}^i represents the i^{th} perturbed observation ensemble member at time $t + 1$ and \mathbf{v}_{t+1}^i indicates the associated perturbation. The unbiased

empirical estimate \hat{R}_{t+1} of the observation error covariance matrix at time $t+1$ is then obtained with the help of hypothesis (32) by:

$$\hat{R}_{t+1} = \frac{1}{N-1} \sum_{i=1}^N (\mathbf{v}_{t+1}^i) (\mathbf{v}_{t+1}^i)^t \quad (38)$$

In the limit of an infinite ensemble, this matrix converges towards the prescribed error covariance matrix R .

The correction step consisted in updating each forecast state ensemble member by using the current observation. For that, the following linear correction equation is applied:

$$\mathbf{x}_{t+1}^{a,i} = \mathbf{x}_{t+1}^{f,i} + K_{t+1} (\mathbf{y}_{t+1}^i - \mathbf{y}_{t+1}^{f,i}), \quad i = 1, \dots, N \quad (39)$$

The vector $\mathbf{x}_{t+1}^{a,i}$ represents the i^{th} analysis state ensemble member at time $t+1$. The Kalman gain K_{t+1} is expressed starting from ensemble covariance matrices as follows:

$$K_{t+1} = \hat{P}_{xy,t+1}^f \left(\hat{P}_{yy,t+1}^f + \hat{R}_{t+1} \right)^{-1} \quad (40)$$

In this expression, $\hat{P}_{xy,t+1}^f$ indicates the unbiased empirical estimate of the forecast cross error covariance matrix of state and output at time $t+1$. Similarly, $\hat{P}_{yy,t+1}^f$ represents the unbiased empirical estimate of the forecast output error covariance matrix at time $t+1$. These estimates are obtained by following a similar protocol to the one used to compute \hat{P}_{t+1}^f . In this manner, by introducing $\bar{\mathbf{y}}_{t+1}^f$ the empirical mean of the forecast output ensemble at time $t+1$, we obtain:

$$\hat{P}_{xy,t+1}^f = \frac{1}{N-1} \sum_{i=1}^N \left(\mathbf{x}_{t+1}^{f,i} - \bar{\mathbf{x}}_{t+1}^f \right) \left(\mathbf{y}_{t+1}^{f,i} - \bar{\mathbf{y}}_{t+1}^f \right)^t \quad (41)$$

$$\hat{P}_{yy,t+1}^f = \frac{1}{N-1} \sum_{i=1}^N \left(\mathbf{y}_{t+1}^{f,i} - \bar{\mathbf{y}}_{t+1}^f \right) \left(\mathbf{y}_{t+1}^{f,i} - \bar{\mathbf{y}}_{t+1}^f \right)^t \quad (42)$$

If the number of measurements is larger than the number of ensemble members, the matrix to be inversed in (40) may become singular. In this case, Evensen recommends to use a pseudo inversion based on singular value decomposition

[10]. Besides, the standard formulation of the Kalman gain can be recovered by linearizing the operator h which indirectly appears in relation (40). However, the advantages of this representation are that no linearization is required and there is no need to propagate the forecast error covariance matrix. This technique considerably reduces the computational cost and save storage capacity [36]. Moreover, since each ensemble member evolves in time independently, the ensemble Kalman filter is particularly well suited for parallel processing.

4.3. Dual state-parameter estimation with EnKF

At this point, we have supposed to entirely know the vector of parameters which appears in the nonlinear dynamic system (29). In practice, most of these parameters may be unknown or imprecise. The filtering method had to preserve good convergence properties when process state and model parameters were simultaneously estimated from available measurements. There are two ways to satisfy this requirement.

The first method proceeds by joint estimation and consists in augmenting the unknown state vector by the unknown vector of parameters. The model error covariance matrix Q then synthesizes the uncertainty associated with state and parameters. It is generally recommended to reduce the confidence level of the forecast parameters to maximize the impact of measurements during the update process. However, to prevent manipulating a high-dimensional augmented state vector when the number of parameters is excessive, we prefer avoiding this configuration and using the second method which is more appropriate.

This approach consists in combining two filters in order to dually estimate, alternatively and separately at each iteration, the model parameters from the previous updated state, then the new state from the parameters freshly analyzed. The vector of parameters is estimated similarly to the state vector, but a forced random walk is specified for the parameters, where the proposal distribution $\boldsymbol{\eta}_t$ is Gaussian with zero mean and covariance matrix equal to Z :

$$\boldsymbol{\eta}_t \sim \mathcal{N}(0, Z) \quad (43)$$

In this manner, the parameter prediction step becomes:

$$\boldsymbol{\theta}_{t+1}^{f,i} = \boldsymbol{\theta}_t^{a,i} + \boldsymbol{\eta}_t^i \quad (44)$$

The vector $\boldsymbol{\theta}_{t+1}^{f,i}$ represents the i^{th} forecast parameter ensemble member at time $t + 1$ and $\boldsymbol{\eta}_t^i$ indicates the associated additive noise. Similarly, the vector $\boldsymbol{\theta}_t^{a,i}$ corresponds to the i^{th} analysis parameter ensemble member at time t . The forecast parameter ensemble are then updated from the available measurements. For that, the forecast state ensemble at time $t + 1$ is firstly carried out from the forecast parameter ensemble at time $t + 1$ and from the analysis state ensemble at time t . Thus:

$$\mathbf{x}_{t+1}^{f,i} = g\left(\mathbf{x}_t^{a,i}, \boldsymbol{\theta}_{t+1}^{f,i}, \mathcal{U}_t\right), \quad i = 1, \dots, N \quad (45)$$

The forecast output ensemble is secondly obtained by:

$$\mathbf{y}_{t+1}^{f,i} = h\left(\mathbf{x}_{t+1}^{f,i}\right), \quad i = 1, \dots, N \quad (46)$$

By remembering that we have to perturb observations (*cf.* equation (37)) and to empirically estimate the measurement error covariance matrix (*cf.* equation (38)), the forecast parameter ensemble was updated with the following Kalman analysis step:

$$\boldsymbol{\theta}_{t+1}^{a,i} = \boldsymbol{\theta}_{t+1}^{f,i} + K1_{t+1} \left(\mathbf{y}_{t+1}^i - \mathbf{y}_{t+1}^{f,i}\right), \quad i = 1, \dots, N \quad (47)$$

The Kalman gain, which is used to correct the parameter trajectory, was computed using the following equation:

$$K1_{t+1} = \widehat{P}_{\theta y, t+1}^f \left(\widehat{P}_{yy, t+1}^f + \widehat{R}_{t+1}\right)^{-1} \quad (48)$$

In this relation, $\widehat{P}_{\theta y, t+1}^f$ indicates the unbiased empirical estimate of the forecast cross error covariance matrix of parameters and output at time $t + 1$.

Once the forecast parameter ensemble has been updated, the second filter took over to estimate the state of the system at time $t + 1$ by supposing that the model parameters were known. In other words, we granted our confidence

to the parameter ensemble recently updated by the first filter. The prediction-correction mechanism was then similar to the classical EnKF one. First, the forecast state ensemble at time $t + 1$ was carried out from the analysis state ensemble at time t and from the analysis parameter ensemble at time $t + 1$:

$$\mathbf{x}_{t+1}^{f,i} = g\left(\mathbf{x}_t^{a,i}, \boldsymbol{\theta}_{t+1}^{a,i}, \mathcal{U}_t\right) + \mathbf{w}_t^i, \quad i = 1, \dots, N \quad (49)$$

Second, the forecast output ensemble is then calculated by:

$$\mathbf{y}_{t+1}^{f,i} = h\left(\mathbf{x}_{t+1}^{f,i}\right), \quad i = 1, \dots, N \quad (50)$$

Then, the observations are once more perturbed and we empirically estimate the measurement error covariance matrix. From then on, the forecast state ensemble was updated with the following Kalman analysis step:

$$\mathbf{x}_{t+1}^{a,i} = \mathbf{x}_{t+1}^{f,i} + K_{t+1}\left(\mathbf{y}_{t+1}^i - \mathbf{y}_{t+1}^{f,i}\right), \quad i = 1, \dots, N \quad (51)$$

The Kalman gain, which is used to correct the state trajectory, is computed as previously (*cf.* equation (40)). Let us underline that the prediction-correction procedure for approximating the state of the system can precede the one which operates on the model parameters without any consequence on the estimation quality. If the command input is unknown, we can estimate it by applying the same procedure than the one used for parameter estimation, *i.e.* by associating to the input command a proper uncertainty.

4.4. Sensor network deployment strategy for dual state-parameter estimation

The estimation of both parameters and state variables on a large grid often leads to the classical observation problem. Indeed, a reduction of the sensor number significantly deteriorates estimation-quality [38]. This loss of precision results from the lack of available information to estimate the state. In the case of a linear spatio-temporal dispersion model, the rank of the observation matrix decreases if the number of sensors is low [39]. Moreover, by using a single mobile sensor, the rank of the observation matrix also decreases when the distance between the sensor and the origin of the flux grows up. Some

works on the network tolerance to loss of sensors, in the context of dynamic system surveillance, have allowed to analyze the observability in function of the set of sensors which are intact [40]. In the nonlinear case, the largest of the eigenvalues of the observability gramian, *i.e.* the largest of the eigenvalues of the positive semi-defined matrix which is solution of the Lyapunov equation, is related to the estimation quality of the considered filter [41]. This quality can also be measured by evaluating the estimation error variance as criterion [38]. Nevertheless, these criteria do not take into account the spatial properties of the estimation quality. Indeed, although it is possible to detect the sensors whose loss would significantly degrade the precision of estimation, no cartography of uncertainties is established. In other words, there is no clue to indicate the optimal sensor location. We propose to adapt the observatory in function of the cell-by-cell output estimation error calculations. Thus we hope to complete the lack of spatial information of a sparse-observation network by adding a mobile sensor. This mobile sensor was then in charge of filling in the missing information by collecting measures with a well-defined moving strategy. At each observation time, the cell-by-cell output estimation error, *i.e.* the diagonal of the analysis output error covariance matrix $\hat{P}_{yy,t+1}^a$, was evaluated. This matrix was obtained by firstly computing the analysis output ensemble at time $t + 1$ from the analysis state ensemble at time $t + 1$:

$$\mathbf{y}_{t+1}^{a,i} = h\left(\mathbf{x}_{t+1}^{a,i}\right), \quad i = 1, \dots, N \quad (52)$$

From then on, by introducing $\bar{\mathbf{y}}_{t+1}^a$ the empirical mean of the analysis output ensemble at time $t + 1$, we were able to evaluate the analysis output error covariance matrix at time $t + 1$ as follow:

$$\hat{P}_{yy,t+1}^a = \frac{1}{N-1} \sum_{i=1}^N \left(\mathbf{y}_{t+1}^{a,i} - \bar{\mathbf{y}}_{t+1}^a\right) \left(\mathbf{y}_{t+1}^{a,i} - \bar{\mathbf{y}}_{t+1}^a\right)^t \quad (53)$$

Note that strictly, we distinguished at each lattice site two estimation error variances which respectively corresponded to the x and y component of the fluid flow speed. The idea consisted in defining a uncertainty function which depended on these variances. The mobile sensor then moved to the position

where the value of this function was high. In other words, the mobile sensor was sent to the position where uncertainty was the highest to collect the relevant information. Once the target site had been designated, the sensor was assigned to move continuously in space. Nevertheless, this approach required the use of obstacle avoidance algorithms in presence of buildings and other obstacles. Moreover, in the case of urban area measurements obtained from instrumented vehicle, only on road movements were enabled. As a result, several iterations were required to reach the target position and an optimal routing algorithm had to be defined. New data can also be collected while the sensor is moving if the technology is available. However, we preferred avoiding this type of movement by means of simplicity. Indeed, our goal being to firstly validate the filtering method applied to the lattice Boltzmann model, we choose a strategy which does not impose any constraint on movement. Consequently, the mobile sensor is instantaneously positioned at the lattice cell where uncertainty is the highest, after a period of time during which some measurements had been carried out. This supposed that the mobile sensor moved by successive jumps from cells to cells when the consecutive highest variances were spatially spread. Obviously, this procedure was not realistic. The aim consisted in quasi-instantaneously completing the lack of available information and studying the influence in terms of estimation quality.

4.5. Semi-physical dynamic modeling

Introduced in the case of artificial neural networks, the semi-physical modeling consists in combining the flexibility of a behavior model implemented by learning mechanisms with the legibility of a knowledge-based model [42, 43]. Indeed, very frequently, and especially in manufacturing, it is inconceivable to find a knowledge-based (or white-box) model which is satisfactory for the purpose of interest (*i.e.* with low computational cost while maintaining high accuracy). The traditional alternative is to design a behavior (or black-box) model obtained from data alone, through an elaborate parameter estimation process. In such a case, the estimated parameters have no physical meaning and cannot

be interpreted. In other words, the model designer does not take into account the hypothetical prior knowledge available as algebraic or differential equations. Consequently, in response to this under-utilization of knowledge, the idea of taking advantage of the existing knowledge while keeping the flexibility of parameterized models trained from data was then introduced under the name of semi-physical (or gray-box) modeling.

A general methodology for designing semi-physical models was proposed in [42]. The first step consisted in implementing as discrete-time neural network with fixed weights the functions which known reliably. To this end, the authors emphasized the importance of the discretization scheme used to transform differential equations arising from physics into discrete time equations that were suitable for numerical processing. Indeed, the stability of the recurrent network model largely depends on the discretization scheme. Obviously, it is premature to consider this model as a gray-box model. At this stage of development, there are several ways of passing to a semi-physical form depending on the confidence level which is related to the previous knowledge-based model. In the case where the value of a parameter is unknown or imprecise, a network training procedure based on experimental data may be set up. The simplest semi-physical model with a single adjustable parameter is then obtained. A second level of criticism towards the model may be considered by adding one hidden layer with non-linear activation function. This more elaborate semi-physical model with new adjustable synaptic weights generally reduces the modeling mean-square error (MSE) on the test sequence. Finally, the dependence between two or more variables may be supposed. This third level of criticism increases the number of hidden layers and usually decreases the modeling MSE if the assumption turns out to be true. In summary, the design process of semi-physical modeling consists in introducing new degrees of freedom (relaxation step) under some considerations.

Under its original form, the lattice Boltzmann scheme is derived from the discretized Boltzmann transport equation. Consequently, it is a knowledge-based model for the kinetic theory of gases. With the aim of defining a semi-physical

LBM (and carrying out an online estimation), we have to introduce adjustable parameters (degrees of freedom) which act for the intrinsic memory of the model like the synaptic weights of a neural network. For that, a level of criticism towards the model may be considered. In particular, let us consider the case of turbulent flows, which may lead to numerical instabilities due to a small kinematic viscosity. As we discussed in section 2, the most popular solution consists in adopting the Smagorinsky subgrid model which locally increase the value of the kinematic viscosity. Now, let us bear in mind that in the MRT approach, the value of the kinematic viscosity is dependent on the relaxation parameters s_7 and s_8 . Hence, by inspiring from the Smagorinsky approach, we suppose that s_7 and s_8 spatially differ. From these new degrees of freedom emerges an elaborate semi-physical model. We hope to decrease the mean-square error and compensate imprecisions such as discretization errors or approximations.

5. Application to the lid-driven cavity flow

5.1. Motivations

We consider the standard lid-driven cavity problem, which involves isothermal and incompressible flow in a two-dimensional square domain. In this specific configuration, all the boundaries of the square are composed of solid cells. The top wall moves with constant velocity tangent to the side (non-homogeneous Dirichlet boundary condition), while the other three are stationary (homogeneous Dirichlet boundary condition). Although this configuration seems to be restrictive, we choose it for several reasons. The lid-driven cavity problem is a typical case of a physical situation which allows us to modelize transverse velocity profiles within an urban street canyon. The term street canyon refers to a relatively narrow street between buildings which line up continuously along both sides, *i.e.* the most frequent configuration observed in urban areas. When the wind direction above roof-top height is perpendicular to the road direction, a vortex circulation appears and dispersion cannot occur since air and consequently air pollutants such as vehicle exhaust pollutants, are trapped within

the street canyon, raising the concentration of this contaminants. Moreover, the lid-driven cavity problem has long been used to test or validate new methods.

5.2. Simulation description

Some precisions concerning the simulation parameters have to be given. Firstly, table 1 reports the different values of the parameters involved in the lattice Boltzmann model for simulating lid-driven cavity flow. These parameters are defined such that the considered fluid corresponds to the ambient air. It constitutes, except contrary mention, the default parameters for all the tests. The Reynolds number associated with this flow is in the range of 10^5 .

Numerical data	
Characteristic length	$L = 1 \text{ m}$
Top wall velocity	$u_x = 1 \text{ m.s}^{-1}$
Kinematic viscosity	$\nu = 1.7 \times 10^{-5} \text{ m}^2.\text{s}^{-1}$
Mean density	$\rho_0 = 1.204 \text{ kg.m}^{-3}$
Mach number	$Ma = 0.1$
Number of nodes	$N_x = 32$
	$N_y = 32$
Relaxation parameters	$s_1 = 1.1$
	$s_2 = 1.1$
	$s_4 = 1.1$
	$s_6 = 1.1$

Table 1: *Simulation parameters in the case of lid-driven cavity flow*

Consequently, this situation involves turbulent flow, for which the Smagorinsky subgrid model was used for stability reasons [16, 17, 4]. The value of the Smagorinsky constant was arbitrarily chosen such that $C_{Smago} = 0.2$ for all the tests. In the lattice Boltzmann model, the time step δ_t was determined with respect to the physical parameters in order to satisfy the stability condition of the explicit Euler scheme. But, this value is often largely smaller than the sampling

cadence of most common sensors. Thus, we introduced for the observations another time step $\delta_{t'}$ such that $\delta_t \ll \delta_{t'}$. This means that the filter works in pure prediction between two observation times. We imposed $\delta_{t'} = 10 \delta_t$ and except contrary mention, 2000 observations are regularly distributed in the time interval, leading to a total simulation time of about 35 s. These observations were simulated by using a classical protocol. The first step consisted in defining \mathbf{x}_0 the reference initial state of the system. Here, \mathbf{x}_0 is the vector of equilibrium moments over the entire domain, which was computed by considering the ambient air density and a null fluid velocity. The explicit Euler scheme was then applied with the time step δ_t and the nonlinear vector function h provided the reference output to estimate. In order to generate the observation sequence, the last step consisted in adding a measurement noise with zero mean and covariance matrix R at each observation time, *i.e.* every 10 time step. Obviously, the vectors of this sequence were not totally exploited during the estimation, since only information at sensor location were required in the filtering process.

5.3. Results

5.3.1. Impact of sensor network deployment strategy

First, we wished to experiment the impact of sensor network deployment strategy on the estimation quality. For that, we consider a particular situation in which the parameters of the GLBE were all known in order to minimize the number of degrees of freedom. On the other hand, we suppose that only a lack of information concerned the input command. In other words, the velocity of the top wall was solely unknown. The x and y components of this velocity were then treated as parameters and had to be estimated. We then defined a simple semi-physical model with two adjustable parameters. We imposed 50 Monte Carlo samples per iteration. Let us bear in mind that a forced random walk is specified for the parameters, where the proposal distribution is Gaussian with zero mean and covariance matrix $Z = 1.0 \times 10^{-8} I_2$. We considered an initial state vector which differs from the reference initial state vector by adding a random Gaussian noise with zero mean and covariance matrix $Q_0 =$

$1.0 \times 10^{-4} I_{9N_n}$. Here, $N_n = N_x \times N_y$ is the total number of nodes. Similarly, the initial input command is randomly generated by adding a random Gaussian noise with zero mean and covariance matrix $Z_0 = 1.0 \times 10^{-3} I_2$ to the reference solution. Concerning measurements, the observation error covariance matrix was set to $R = 1.0 \times 10^{-7} I_{2N_n}$. From then on, we proposed to estimate the signal-to-noise ratio in order to evaluate measurement quality. However, the spatio-temporal behavior of this kind of dynamic fluid flow system required the definition of a signal-to-noise ratio for each node. Thus, we are able to establish a sort of cartography of measurement quality. In the present case, the maximum value of the signal-to-noise ratio is obtained where the norm of the velocity is the highest. On the other hand, if the norm of the velocity was weak, this ratio decreased and sometimes even became negative *i.e.* the signal was completely buried in noise. The mean value of these signal-to-noise ratios over the entire domain borders 20 dB. Finally, the model error covariance matrix was set to $Q = 1.0 \times 10^{-7} I_{9N_n}$ so that the same confidence level was assigned to both model prediction and observations.

As a reference, figure 4 illustrates the norm of the fluid velocity and the streamline contours (with direction arrows) over the entire domain at the final instant. These values were directly computed from the reference output and had to be estimated after 2000 observations. Concerning dynamic flow, we observe the apparition of a major vortex in the center of the cavity, which circulates in clockwise direction. We notice two minor vortices at the inferior corners of the cavity, which circulate in counter clockwise direction. It is at the level of these vortices that vehicles exhaust pollutants can be trapped.

From then on, we count 4 experiments which only differ by the observatory network.

Experiment 1

The observatory network consists of nine sensors which are randomly positioned using a uniform Gaussian distribution. In other words, no information is exploited to optimize sensor location. These sensors are then in charge of collecting information at each observation time and remain fixed during the whole

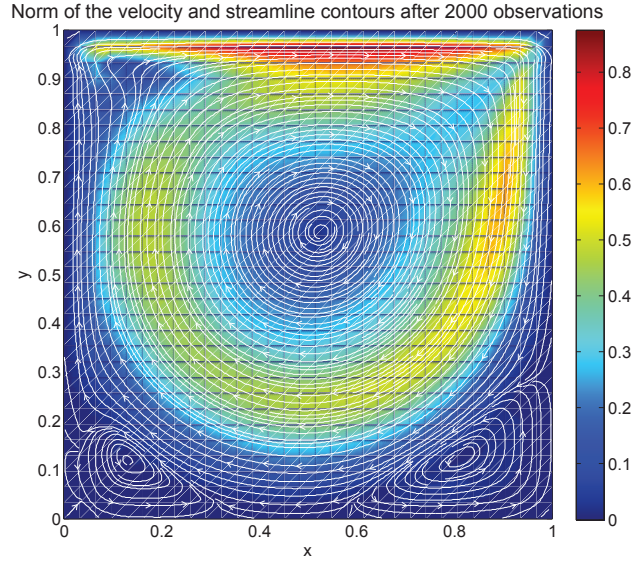


Figure 4: *Reference norm of the fluid velocity and streamline contours at the final instant.*

estimation procedure. Figure 5 illustrates the estimation of the input command vector. We remind that this vector is composed by the x and y components of the velocity of the top wall. The y components of the estimated input command

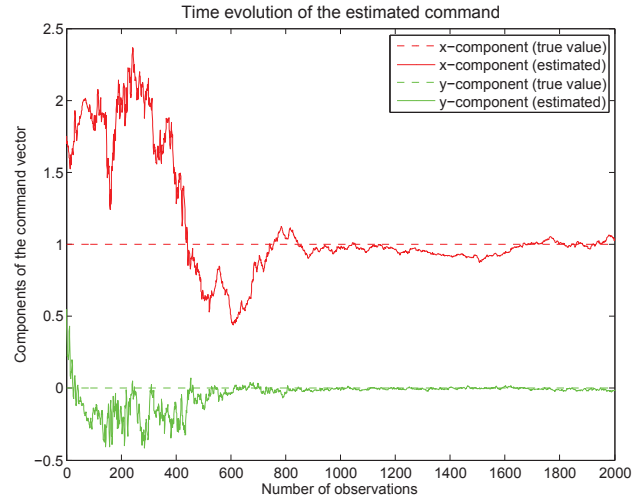


Figure 5: *Estimation of the input command vector components (Experiment 1).*

vector converge to its reference value after about 800 observations, while the x components requires two times more iterations due to the lack of information inherent to random sensor placement. The estimation quality is studied by computing the mean-square error between the estimated and the reference outputs at each observation time. The MSE rapidly but quite irregularly decreases and then stabilizes around 900 observations. We observe a MSE of about 2.0×10^{-4} , which is more than acceptable considering the fact that no *a priori* information has been exploited to improve sensor placement.

Experiment 2

This time the observatory network consists of one single mobile sensor which moves to the position where the output estimation error is highest, with the hope to complete the lack of information. Figure 6 illustrates the estimation of the input command vector. Both components of the estimated input command

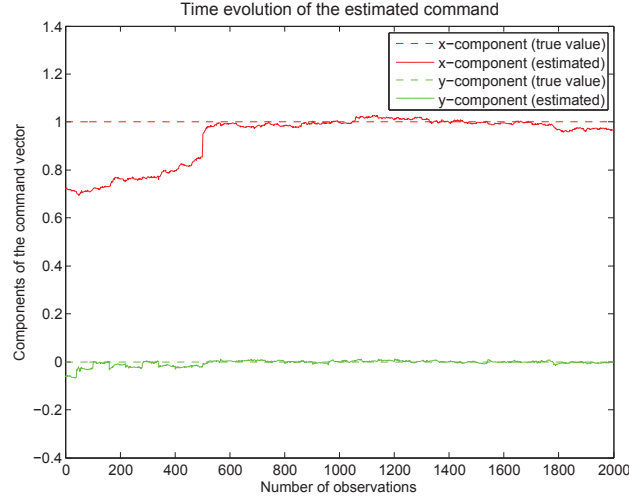


Figure 6: *Estimation of the input command vector components (Experiment 2).*

vector converge to their reference value after about 700 observations. Hence, despite the fact that only one sensor is available, the estimation of the input command vector components is faster than in the previous case. Moreover, the estimation quality of these components is also clearly better. The strategy

which consists in sending the mobile sensor to the sites where the variance of the output error is the highest, tends to be particularly efficient to estimate the input command. The estimation quality is studied as previously by computing the MSE between the estimated and the reference outputs at each observation time. The MSE decreases more rapidly and in a more regular manner than in the previous experiment. Indeed, the error stabilizes around 700 observations. We observe a mean-square error of about 1.0×10^{-4} , *i.e.* slightly better than those obtained with nine times more sensors. The reduction of the number of sensors is largely compensated by introducing one mobile sensor which moves strategically within the area.

To conclude this experiment, we want to detect the locations where the sensor has mainly moved to. The idea being to use this information in order to ideally position fixed sensors and to avoid the utilization of (expensive) mobile sensors. Figure 7 gives an overview of the different locations where the sensor has moved to during the estimation procedure. The path frequencies at each node are illustrated by vertical bar graphs. To facilitate spatial visualization, we represent with the same color all the sites which are located on the same vertical line. The mobile sensor moves priorily towards the two inferior corners of the cavity. On the other hand, the center of the cavity is totally ignored. We notice that the sensor also moves towards the right upper corner, but with a less important path frequency. Thus, the positions where uncertainty is the highest are located at the level of the corners of the domain, and especially where the norm of the velocity is the smallest.

Experiment 3

This experiment is set up to consolidate the sensor network deployment strategy previously exposed. Once again, we considered a single mobile sensor, but this time the target sites are randomly positioned using a uniform Gaussian distribution. Figure 8 gives an overview of the different locations where the sensor has moved to during the estimation procedure. Without surprise, those locations was spread out over the entire domain and path frequencies do not exceed the unit value, since the distribution is uniform. Figure 9 illustrates

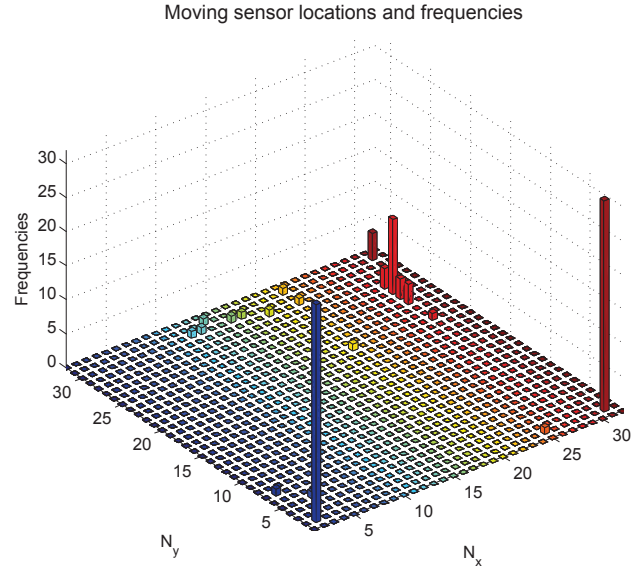


Figure 7: *Mobile sensor locations and path frequencies (Experiment 2).*

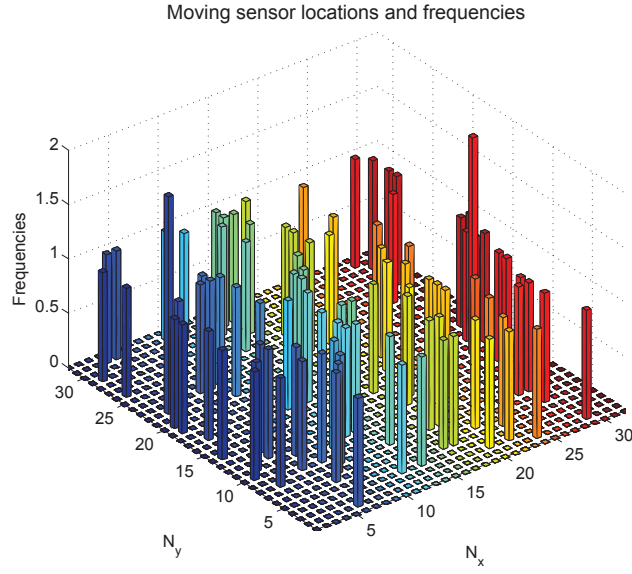


Figure 8: *Mobile sensor locations and path frequencies (Experiment 3).*

the estimation of the input command vector. We immediately observe that both estimated components take more time to converge than in the case where

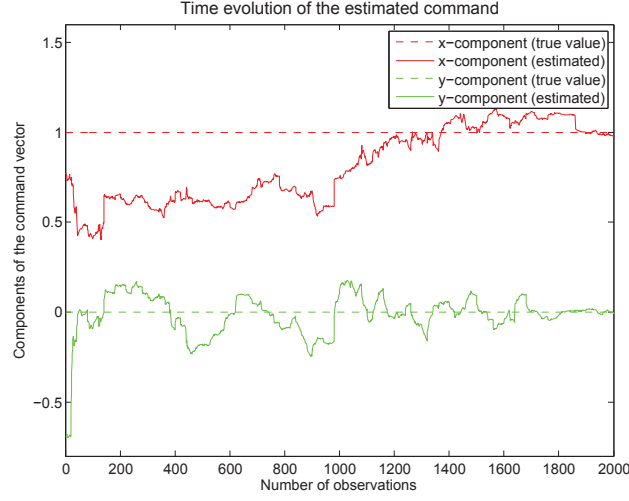


Figure 9: *Estimation of the input command vector components (Experiment 3).*

the sensor moves strategically, and show random fluctuations around the expected value. This is a direct consequence of the fact that the sensor network motion strategy is purely random. Moreover, the mean-square error between the estimated and the reference outputs slowly and irregularly decreases, and then lately stabilizes. We observe a MSE of about 1.0×10^{-3} , *i.e.* ten times larger than those obtained when the sensor moved strategically. Nevertheless, these results remain suitable since the mobile sensor has sometimes moved to the zones of interest. This means that collecting information on some sites is simply useless since it did not improve the estimation quality in return. This remark is a key point in sensor network motion strategy. The expert has to define an optimal strategy in order to avoid any unprofitable motion.

Experiment 4

We exploit the information previously collected concerning the locations of interesting zones. For that, we detected the locations which have been mostly visited by the mobile sensor during the experiment 2. The idea then consisted in deploying four fixed sensors on these particular positions. Figure 10 exhibits the estimation of the input command vector. Both components of the estimated

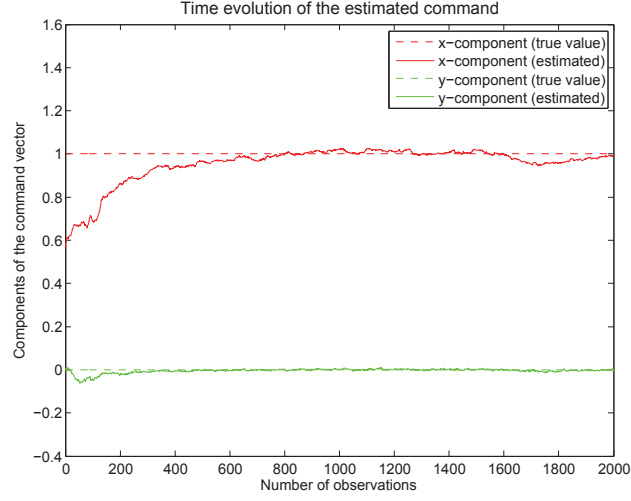


Figure 10: *Estimation of the input command vector components (Experiment 4).*

input command vector converge to their reference value after about 700 observations (against 800 for the experiment 1). Thus, although the number of sensors is almost half reduced, the estimation of the input command vector components is faster and significantly better than in the experiment 1. The estimation quality is studied as previously by computing the mean-square error between the estimated and the reference outputs at each observation time. The MSE quite regularly decreases and then stabilizes around 800 observations (against 900 for the experiment 1). We observe a MSE of about 1.2×10^{-4} (against 2.0×10^{-4} for the experiment 1). Consequently, the strategy which consisted in deploying some fixed sensors on the zones of interest provided better performances in terms of convergence speed and estimation quality, even if there was half as many sensors.

5.3.2. *Impact of semi-physical modeling*

So far, we have supposed to precisely know all the parameters except the input command vector components. However, this was a hypothetical consideration which nearly never occurred in practice. Indeed, even if most of these parameters have physical relevance, they generally remain imprecise or not mea-

surable. Moreover, those which had no real physical meaning were generally empirically determined. In particular, it was the case of the Smagorinsky constant which enables to perform large-eddy simulation of turbulent flows. As a result, the notion of semi-physical modeling which integrated an estimation procedure based on experimental data, revealed to be quite useful. We proposed to add new degrees of freedom to the lattice Boltzmann model to create a more elaborate adaptive (with respect to local effects) semi-physical model. This approach is detailed in experiment 5.

Experiment 5

In this experiment, we consider that the relaxation parameters s_7 and s_8 are spatially heterogeneous. In other words, we supposed that the value of these parameters can spatially differ. The aim of this approach was to bypass the Smagorinsky subgrid model which is crucial for turbulent flow simulations. Indeed, in order to avoid the apparition of numerical instabilities, this method consisted in locally increasing the value of the kinematic viscosity. Now, let us bear in mind that in the MRT approach, the value of the kinematic viscosity is dependent on the relaxation parameters s_7 and s_8 . Consequently, these two relaxation parameters were set to be locally free and an estimation procedure was established, without applying the Smagorinsky subgrid method. From these new $2N_n$ degrees of freedom emerges an elaborate semi-physical model which was much more adaptive than the previous one. Let us also remember the equality constraint $s_7 = s_8$ in order to enforce isotropy. Hence, we counted as many additional degrees of freedom as there are cells in the domain. Once again, we proposed to estimate these parameters by using the dual state-parameter estimation procedure. For that, a forced random walk was specified, where the proposal distribution is Gaussian with zero mean and covariance matrix $Z = 1.0 \times 10^{-5} I_{N_n}$. We deployed four fixed sensors on the same particular zones of interest of experiment 4. In the case where new zones of interest due to extra parameters appeared, we additionally configured a mobile sensor which moved to the position where uncertainty is the highest to collect the relevant information. Finally, we consider a longer simulation time than previously, since

sometimes the parameter convergence tend to be very slow. For that, we always imposed the condition $\delta_{t'} = 10\delta_t$, but this time 10000 observations was regularly distributed in the time interval. Considering the large amount of parameters involved, we expected a longer convergence time than previously. Figure 11 gives an overview of the numerical values of s_7 obtained at each lattice site at the final instant by using the standard Smagorinsky subgrid method. We observe

s_7 obtained by the Smagorinsky subgrid method at each site after 10000 observations

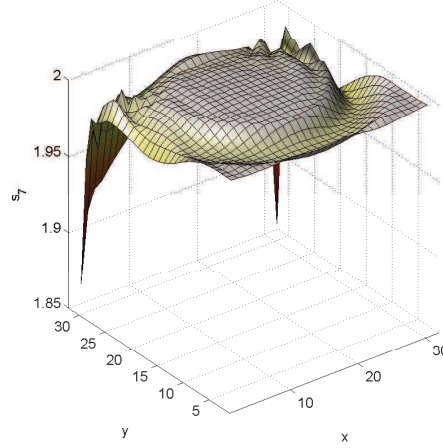


Figure 11: Numerical values of the relaxation parameters s_7 obtained at each lattice site by the Smagorinsky subgrid method at the final instant (Experiment 5).

that the numerical values of s_7 preserve a sort of spatial homogeneity and more or less graphically represent the behavior of the dynamic fluid flow system. We distinguish two lattice sites where the numerical values of s_7 significantly differ from the other ones. These sites are located at the superior corners of the cavity. Figure 12 represents the estimated values of s_7 at each lattice site obtained by the dual EnKF at the final instant.

At first sight this estimation would appear to be incorrect since the estimated values of s_7 totally differ from those obtained by the Smagorinsky subgrid method. Besides, some parameters even came close to be null, which seems to be physically absurd. We suspected that the large number of parameters to optimize involves a sort of restriction. Nevertheless, the results exhibited some

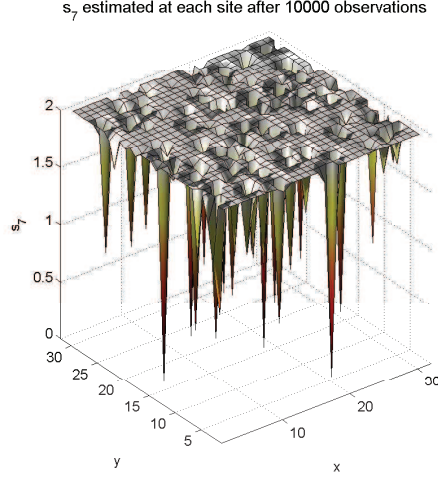


Figure 12: *Estimated values of the relaxation parameters s_7 obtained at each lattice site by the dual EnKF at the final instant (Experiment 5).*

interesting properties. First, concerning flow dynamics, we clearly distinguish the major vortex in the center of the cavity and even the apparition of the two minor vortices at the bottom corners of the cavity, which circulate in counter clockwise direction. Then, the estimated value of the fluid flow velocity is globally correct, whereas the contour plot of the flow field shows some unwelcome activity near the boundaries and the secondary vortices. These results are illustrated in figure 13. Finally, the MSE between the estimated and the reference outputs slightly decreases at each observation and stabilizes to a value of about 2.0×10^{-3} . All these results clearly denote the absence of numerical instabilities that would have theoretically occurred. Indeed, the Reynolds number is above the value at which turbulent flow is produced and no subgrid method has been specified. The key point behind all this is that the physical equations, which appear in the Smagorinsky subgrid method, have been taken out from the model. Consequently, the kinematic viscosity has to be locally modified without any physical consideration. That is the reason why the adjustable relaxation parameters, which have been introduced as degrees of freedom in the semi-physical lattice Boltzmann model and adjusted during the filtering proce-

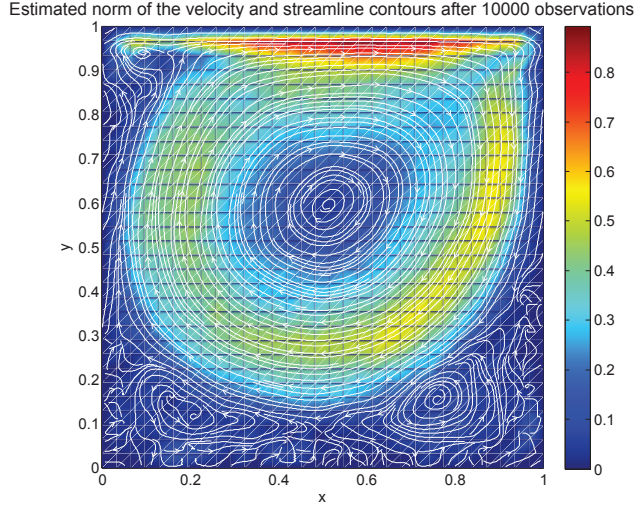


Figure 13: *Estimated norm of the fluid velocity and streamline contours at the final instant (Experiment 5).*

dure, have no physical meaning. However, as we have noticed, the performances in terms of estimation quality of this filtering procedure based on experimental data were completely suitable. The high amount of activity at, or in the vicinity of, boundaries and corners is due to the lack of information (limited number of sensors) and to the difficulty to estimate low velocities, especially in presence of dominant measurement noise. Thus, we perceive the effects of the semi-physical modeling, which consists in increasing the estimation quality, or at least, avoiding the apparition of numerical instabilities, by leaving free some parameters which have no physical meaning. A major approximation introduced in EnKF is related to the use of a limited number of ensemble realizations. The ensemble size limits the space where the solution is searched for and in addition introduces spurious correlations that lead to excessive decrease of the ensemble variance and possibly divergence. The spurious correlations can be handled by localization methods that attempt to reduce the impact of measurements that are located far from the grid-point to be up-dated. Localization methods either filter away distant measurements or attempt to reduce the amplitude of the

long-range spurious correlations. The use of a local analysis scheme effectively increases the ensemble solution space while reducing the impact of spurious correlations. The use of local analysis scheme allows for a relatively small ensemble size to be used with a high-dimensionnal dynamical model.

6. Conclusion

In this paper, we presented some significant advantages of introducing the concept of semi-physical modeling in the lattice Boltzmann approach. Indeed, by associating this principle with the dual state-parameter estimation procedure, we showed that the new model is able to adapt to different kinds of driving behavior, as long as there are measurement data available. We succeeded in estimating a sequence of parameters with or without physical meaning and in performing stable LES (Large Eddy Simulation) of turbulent flows without using the Smagorinsky subgrid method. However, this estimation tended to be extremely slow and the convergence time had to be taken into account, before operational use, for monitoring the system. We also observed a loss of the physical meaning of some relaxation parameters. In order to avoid this loss, we envisage as future work to restrict the spatial heterogeneity hypothesis to some particular zones of interest, *i.e.* to reduce the number of degrees of freedom.

We also experimentally observed that the performances of the filtering procedure in terms of convergence speed and estimation quality were largely dependent on sensor location, no matter what their number was. We noted that collecting information at some sites of the area does not have any impact, or few, on the estimation quality. On the other hand, we showed that some particular sites constitute the zones of interest, where it was crucial to send a mobile sensor during some laps of time or to deploy a fixed sensor, to ensure that the filter works correctly. We proposed a strategy to detect these zones of interest, which consisted in sending during the estimation a mobile sensor to the sites where the variance of the output error was maximum. We then considered that the sites which have been mostly visited constitute the zones of interest, and

we succeeded in obtaining excellent performances in terms of convergence speed and estimation quality with a reduced number of sensors.

References

- [1] Y. H. Qian, D. d’Humières, P. Lallemand, Lattice BGK models for Navier-Stokes equation, *Europhysics Letters* 17 (6) (1992) 479-484.
- [2] B. Chopard, A. Dupuis, A. Masselot, P. Luthi, Cellular automata and lattice Boltzmann techniques: an approach to model and simulate complex systems, *Advances in Complex Systems* 5 (2) (2002) 103-246.
- [3] X. He, L.-S. Luo, Lattice Boltzmann model for the incompressible Navier-Stokes equation, *Journal of Statistical Physics* 88 (3-4) (1997) 927-944.
- [4] B. Chopard, M. Droz, Cellular automata modeling of physical systems, Cambridge University Press, 1998.
- [5] S. Succi, The lattice Boltzmann equation for fluid dynamics and beyond, Oxford University Press, 2001.
- [6] D. d’Humières, Generalized lattice Boltzmann equations, in: B. D. Shizgal, D. P. Weaver (Eds.), *Rarefied gas dynamics: theory and simulations*, vol. 159, Progress in Astronautics and Aeronautics, 450-458, 1992.
- [7] P. Lallemand, L.-S. Luo, Theory of the lattice Boltzmann method: dispersion, dissipation, isotropy, Galilean invariance, and stability, *Physical Review E* 61 (6) (2000) 6546-6562.
- [8] G. Evensen, Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, *Journal of Geophysical Research* 99 (C5) (1994) 10143-10162.
- [9] G. Burgers, J. van Leeuwen, G. Evensen, Analysis scheme in the ensemble Kalman filter, *Monthly Weather Review* 126 (6) (1998) 1719-1724.

- [10] G. Evensen, The ensemble Kalman filter: theoretical formulation and practical implementation, *Ocean Dynamics* 53 (4) (2003) 343-367.
- [11] L. Bourgois, G. Roussel, M. Benjelloun, Apprentissage séquentiel d'un schéma de Boltzmann sur réseau pour l'assimilation de champs de vent locaux, in: *Proceedings of the 6th STIC et Environnement Conference*, Calais, France, 2009.
- [12] H. Moradkhani, S. Sorooshian, H. V. Gupta, P. R. Houser, Dual state-parameter estimation of hydrological models using ensemble Kalman filter, *Advances in Water Resources* 28 (2) (2005) 135-147.
- [13] P. L. Bhatnagar, E. P. Gross, M. Krook, A model for collision processes in gases. Part I. Small amplitude processes in charged and neutral one-component systems, *Physical Review* 94 (3) (1954) 511-525.
- [14] A. J. C. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part I. Theoretical foundation, *Journal of Fluid Mechanics* 271 (1994) 285-309.
- [15] A. J. C. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part II. Numerical results, *Journal of Fluid Mechanics* 271 (1994) 311-339.
- [16] S. Hou, J. Sterling, S. Chen, G. D. Doolen, A lattice Boltzmann subgrid model for high Reynolds number flows, *Fields Institution Communications* 6 (1996) 151-166.
- [17] J. Smagorinsky, General circulation experiments with the primitive equations, *Monthly Weather Review* 91 (3) (1963) 99-164.
- [18] D. d'Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, L.-S. Luo, Multiple-relaxation-time lattice Boltzmann models in three dimensions, *Philosophical Transactions of the Royal Society of London A* 360 (1792) (2002) 437-451.

- [19] R. L. Bras, I. Rodriguez-Iturbe, Random functions and hydrology, Dover Publications, 1994.
- [20] D. Q. Zheng, J. K. C. Leung, B. Y. Lee, Online update of model state and parameters of Monte Carlo atmospheric dispersion model by using ensemble Kalman filter, *Atmospheric Environment* 43 (12) (2009) 2005-2011.
- [21] E. Todini, A. Szollosi-Nagy, E. F. Wood, Adaptive state-parameter estimation algorithm for real time hydrologic forecasting: a case study, in: *Proceedings of the IISA/WMO Workshop on the Recent Developments in Real Time Forecasting/Control of Water Resources Systems*, Laxemburg, Austria, 1976.
- [22] E. Todini, Mutually interactive state-parameter (MISP) estimation: application of Kalman filter to hydrology, hydraulic and water resources, in: *Proceedings of the AGU Chapman Conference*, Pittsburg, Pennsylvania, USA, 1978.
- [23] A. H. Jazwinski, *Stochastic processes and filtering theory*, Academic Press, 1970.
- [24] E. A. Wan, A. T. Nelson, Dual Kalman filtering methods for nonlinear prediction, smoothing and estimation, *Advances in Neural Information Processing Systems* 9 (1996) 793-799.
- [25] A. T. Nelson, Nonlinear estimation and modeling of noisy time-series by dual Kalman filtering methods, Ph.D. thesis, Oregon Graduate Institute of Science & Technology, 2000.
- [26] G. Evensen, Using the extended Kalman filter with a multilayer quasi-geostrophic ocean model, *Journal of Geophysical Research* 97 (C11) (1992) 17905-17924.
- [27] R. N. Miller, M. Ghil, F. Gauthiez, Advanced data assimilation in strongly nonlinear dynamical systems, *Journal of the Atmospheric Sciences* 51 (8) (1994) 1037-1056.

- [28] A. Gelb, J. F. Kasper, R. A. Nash, C. F. Price, A. A. Sutherland, Applied optimal estimation, The M.I.T. Press, 1974.
- [29] S. J. Julier, J. K. Uhlmann, A new extension of the Kalman filter to non-linear systems, in: Proceedings of the SPIE 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Orlando, Floride, USA, 1997.
- [30] J. H. Kotecha, P. M. Djuric, Gaussian particle filtering, IEEE Transactions on Signal Processing 51 (10) (2003) 2592-2601.
- [31] R. H. Reichle, D. B. McLaughlin, D. Entekhabi, Hydrologic data assimilation with the ensemble Kalman filter, Monthly Weather Review 130 (1) (2002) 103-114.
- [32] P. R. Houser, R. D. Koster, R. H. Reichle, J. P. Walker, Extended versus ensemble Kalman filtering for land assimilation, Journal of Hydrometeorology 3 (6) (2002) 728-740.
- [33] G. Evensen, Data Assimilation: The Ensemble Kalman Filter (2nd Edition), Springer, 2009.
- [34] A. Stordal, H. Karlsen, G. Naevdal, H. Skaug, Bridging the ensemble Kalman filter and particle filters: the adaptive Gaussian mixture filter, Computational Geosciences 15 (2) (2011) 293-305.
- [35] V. D. Tran, V. Monbet, F. Le Gland, Filtre de Kalman d'ensemble et filtres particuliers pour le modèle de Lorenz, in: Proceedings of the MajecSTIC Conference, Lorient, France, 2006.
- [36] S. Gillijns, O. Barrero Mendoza, J. Chandrasekar, B. L. R. de Moor, D. S. Bernstein, A. Ridley, What is the ensemble Kalman filter and how well does it work?, in: Proceedings of the American Control Conference, Minneapolis, Minnesota, USA, 2006.

- [37] A. Coman, Modélisation spatio-temporelle de la pollution atmosphérique urbaine à partir des mesures d'un réseau de surveillance de la qualité de l'air, Ph.D. thesis, Université de Paris XII, 2008.
- [38] V. Sircoulomb, G. Hoblos, H. Chafouk, J. Ragot, Evaluation de la qualité d'estimation en fonction de la perte de capteurs, Diagnostic des systèmes complexes (2008) 9–26.
- [39] E. Ternisien, Caractérisation aveugle d'un système de dispersion en représentation d'état et localisation de source, Ph.D. thesis, Université du Littoral Côte d'Opale, 2001.
- [40] M. Staroswiecki, G. Hoblos, A. Aitouche, Fault tolerance analysis of sensor systems, in: Proceedings of the 38th IEEE Conference on Decision and Control, Phoenix, Arizona, USA, 1999.
- [41] M. Staroswiecki, On reconfigurability with respect to actuator failures, in: Proceedings of the 15th IFAC World Congress, Barcelona, Spain, 2002.
- [42] Y. Oussar, G. Dreyfus, How to be a gray-box: dynamic semi-physical modeling, Neural Networks 14 (9) (2001) 1161-1172.
- [43] G. Dreyfus, J. M. Martinez, M. Samuelides, M. B. Gordon, F. Badran, S. Thiria, L. Hérault, Réseaux de neurones : méthodologies et applications, Eyrolles, 2004.